TDTS06 – Computer Networks

# Distance Vector Routing

# Assignment 4

# I.Distance Vector Algorithm:

The algorithm we used to find the shortest way from a router to another is the Distance Vector Algorithm. This works as follows: Each node knows the cost to reach each of his neighbor. So, to compute the shortest to any node (y), it takes the minimum between: the cost of each neighbor + their cost to y. So, what is important is that each node only interacts with its neighbors and has no precise knowledge of the overall network.

# II.How we tested our algorithm:

To test our algorithm, first without Poison Reverse, we ran it with the different topologies and made sure that the tables were updated at the right time with the right values, by printing them and analyzing how it behaved when a new node ‘came’ in the network, and what was our final result. Of course, at this point we observed the count-to-infinity problem after the event occurred. Since it was topologies with a small number of nodes, it was easy to check if the values were correct by computing them ourselves and then comparing with what the program produced.

With Poison Reverse, we could notice the change of behavior as it was now shorter, because we didn’t iterate all along.

# III.Case where Poison Reverse fails:

In this topology, if the link between 0 and 3 fails, even with our Poison Reverse, the routing will loop between the 3 other nodes (0, 1, 2) because they will always think the shortest way to 3 is one the others. Because we have more than 2 routers in the loop, it can’t be solved with our solution.

# IV. A way to solve it:

A way to solve it would be to add a piece of information in our tables. If we also add through which node we go through for the next hop, 1 and 2 will realize they both need to go through 0 and wouldn’t decide to go back to the other.